13/5/4 (Item 1 from file: 60)
DIALOG(R)File 60: ANTE: Abstracts in New Tech & Engineer

0000677763 IP Accession No: 2008444186
Multi-node user interface component and method thereof for use in accessing a plurality of linked records

Bates, Cary Lee; Day, Paul Reuben
, USA
Publisher Url: http://patft.uspto.gov/netacgi/nph-Parser?Sect1=PTO2&Sect2=HITOFF&u =/netaht ml/PTO/search-adv.htm&r=1&p=1&f=G&l=50&d=PTXT&S1=58 77766.PN.&OS=pn/5877766& RS=PN/5877766

Document Type: Patent
Record Type: Abstract
Language: English
File Segment: ANTE: Abstracts in New Technologies and Engineering

Abstract:
A user interface component and method of using the same graphically display linked **records** with node display elements representing individual **records**, and optional link display elements representing the links therebetween. The user interface component may be automatically and dynamically generated during navigation between linked **records**, such that whenever a new link is taken from a **record** represented by a node display element, a new node display element is generated. In addition, the user interface component may further be configured to graphically represent the particular location of a particular link within a **record**, as well as a scroll display element associated with selected node display elements such that specific data or locations within the **records** associated therewith may be accessed. Moreover, the user interface component may also be configured to incorporate a retrieve progress display element that indicates a current status of a retrieve operation for a **record**. Individual node display elements **within** a user interface **component** may also **separately** indicate status information for their associated **records**, e.g., displaying a **cache** status and/or a matching status that indicates whether associated **records** match a predetermined search criteria, among others. The user interface component may also be utilized to perform common operations such as printing, caching and loading, among others, on selected **records** in response to user selection of the nodes associated with the selected **records**.

Descriptors: User interfaces; Business machines; Caching; Navigation; Printing

17/5/34 (Item 5 from file: 60)
DIALOG(R)File 60: ANTE: Abstracts in New Tech & Engineer

0000735903 IP Accession No: 2008377773
Program launch acceleration using ram cache

Ballard, Clinton L; Smith, Timothy W
, USA

Publisher Url: http://patft.uspto.gov/netacgi/nph-Parser?Sect1=PTO2&Sect2=HITOFF&u =/netaht ml/PTO/search-adv.htm&r=1&p=1&f=G&l=50&d=PTXT&S1=59 33630.PN.&OS=pn/5933630& RS=PN/5933630

Document Type: Patent
Record Type: Abstract
Language: English
File Segment: ANTE: Abstracts in New Technologies and Engineering

Abstract:

Launch time for a computer program is reduced by logging hard disk accesses during an initial launch, then processing the log file to accelerate subsequent launches. The log file is processed by identifying all the file **portions** accessed during the launch, eliminating any **duplicate** cluster accesses, then sorting the remaining **accesses**. The disk **access log** entries are sorted by physical address or are grouped by file, then organized by logical address within each group. The processed log file is stored with the application program. When the application program is launched thereafter, the processed **log** file is **accessed** first. All the disk **accesses** in the **log** file are performed moving all the data into RAM cache. When the program launch resumes, the launch occurs faster because all the data is already in cache.

Descriptors: Launches; Disks; Applications programs; Computer programs; Acceleration; Reproduction; Logging; Software; Random access memory; Clusters; Sorting

17/5/36 (Item 7 from file: 60)
DIALOG(R)File 60: ANTE: Abstracts in New Tech & Engineer

0000573900 IP Accession No: 2008216053

Apparatus and method for loading and reloading HTML pages having cacheable and non-cacheable portions

Hawes, Michael Kerrigan
, USA

Publisher Url: http://patft.uspto.gov/netacgi/nph-Parser?Sect1=PTO2&Sect2=HITOFF&u =/netaht ml/PTO/search-adv.htm&r=1&p=1&f=G&l=50&d=PTXT&S1=60 61715.PN.&OS=pn/6061715& RS=PN/6061715

Document Type: Patent
Record Type: Abstract
Language: English
File Segment: ANTE: Abstracts in New Technologies and Engineering

Abstract:

On an embedded device with a web server, pages are marked as having non-cacheable HTML portions and cacheable graphics portions. Marking the HTML portion as non-cacheable allows for the retrieval of a web page containing the latest status information without retrieving the graphics images. A refresh function is provided that allows for the comparing of non-**cached portion** time stamps with the stamps of the page on the web server. When the time stamp of the page on the web server is more current than the time stamp of the non-**cached portion**, the non-cacheable **portion** of the web page is retrieved from the web server. Retrieving the non-cacheable HTML **portion** from the web site without retrieving the **cached portion** reduces the amount of time needed to refresh the display image. A timer may be employed to refresh the web page at predetermined intervals, while **using** the current URL or the **history** list of the browser to reach the desired web site and retrieve the current status of the embedded device. The refresh function may be activated by a button available on the browser or by a refresh button on the web page.

Descriptors: World Wide Web; Websites; HTML; Servers (computers); HyperText Markup Language; Images; Buttons; Timing devices; Activated; Retrieval

17/5/37 (Item 8 from file: 60)
DIALOG(R)File 60: ANTE: Abstracts in New Tech & Engineer

0000554505 IP Accession No: 2008204733
Memory management for navigation system

Crowley, Paul; Jaugilas, John; Nash, Alex; Natesan, Senthil; Lampert, David S
, USA
Publisher Url: http://patft.uspto.gov/netacgi/nph-Parser?Sect1=PTO2&Sect2=HITOFF&u =/netaht ml/PTO/search-adv.htm&r=1&p=1&f=G&l=50&d=PTXT&S1=60 73076.PN.&OS=pn/6073076& RS=PN/6073076

Document Type: Patent
Record Type: Abstract
Language: English
File Segment: ANTE: Abstracts in New Technologies and Engineering

Abstract:

A method and system for managing memory resources in a system used in conjunction with a navigation application program that accesses geographic data. The geographic data are comprised of a plurality of data records. The plurality of data records are organized into parcels, each of which contains a portion of the plurality of data records, such that the data records in each portion of the plurality of data **records** that forms each parcel are **accessed** together. One or more buffers each that forms a contiguous **portion** of the memory of the navigation system is provided as a **cache** to store a plurality of parcels. One or more data structures located outside the contiguous **portion** of memory identify the parcels of data stored in the **cache** and the locations in the **cache** at which the parcels are stored. The one or more data structures located outside the contiguous **portion** of memory in which the parcels are **cached** are used to manage the parcel **cache** to use it efficiently. These one or more data structures located outside the contiguous memory in which the parcels are cached are also used to defragment the parcel cache.

Descriptors: Parcels; Data structures; Navigation; Navigation systems; Applications programs; Memory management; Buffers

17/5/40 (Item 11 from file: 60)
DIALOG(R)File 60: ANTE: Abstracts in New Tech & Engineer

0000522472 IP Accession No: 2008141930

Large capacity storage apparatus having storage cells, an accessor, a cache memory and a disc update section to set a number of frequently accessed storage media

Takagi, Shiro
, USA

Publisher Url: http://patft.uspto.gov/netacgi/nph-Parser?Sect1=PTO2&Sect2=HITOFF&u =/netaht ml/PTO/search-adv.htm&r=1&p=1&f=G&l=50&d=PTXT&S1=61 31147.PN.&OS=pn/6131147&RS=PN/6131147

Document Type: Patent
Record Type: Abstract
Language: English
File Segment: ANTE: Abstracts in New Technologies and Engineering

Abstract:

Data is stored in a cache memory, a cache HDD, high frequently accessed optical discs, or low frequently accessed optical discs, and the data is transferred between the cache memory and the cache HDD under the control of a control device which predicts the access frequency of data on the basis of the **access history**, transfers modified data items and only data items frequently accessed and not allocated to a high frequently accessed optical disc from the cache memory to the cache HDD, and deletes the remaining, thereby improving the use efficiency of the cache HDD.

Descriptors: Optical discs; Access control; Discs; Disks; Accessories

Dialog eLink: USPTO Full Text Retrieval Options

17/5/62 (Item 1 from file: 2)
DIALOG(R)File 2: INSPEC

09073568
Title: **Application-level data caching**
Author(s): Boal, P.E.
Journal: Dr. Dobb's Journal , vol.28 , no.12 , pp.30-4
Publisher: CMP Media LLC
Country of Publication: USA
Publication Date: Dec. 2003
ISSN: 1044-789X
SICI: 1044-789X(200312)28:12L.30:ALDC;1-5
CODEN: DDJSDM
Language: English
Document Type: Journal Paper (JP)
Treatment: Practical (P)
Abstract: We present an application-level data-caching library. The first thing to consider when developing an application-level caching library is to decide which features are needed to support client applications. The application-level caching library is encapsulated by a single class called "Lookup" (and internal helper class called "LookupSQL" that generates SQL statements). In general, Lookup **caches** are query/response pairs (key/value pairs) where the query (key) **component** is always unique. Other requirements include: storing dictionary (query/response) type pairs for lookup operations; allowing runtime definition of what set of values needs to be cached; providing standard methods for **accessing** cached values; maintaining **records** of how many requests succeed or fail; providing standard error handling/messaging; supporting requests and caches in various combinations; prefetching memory cache; prefetching disk cache; supporting dynamic memory cache, database lookup, and assignment; single database tables, and freeform SQL queries. The Lookup class supports all of these requirements and can be extended to support other requirements applications might specify
Subfile(s): C (Computing & Control Engineering)
Descriptors: cache storage; query processing; SQL; table lookup
Identifiers: application-level data-caching library; Lookup caches; prefetching memory cache; database lookup; SQL queries; query processing
Classification Codes: C6120 (File organisation); C6130 (Data handling techniques); C6160 (Database management systems (DBMS))
INSPEC Update Issue: 2004-034

17/5/66 (Item 5 from file: 2)
DIALOG(R)File 2: INSPEC

07128236
Title: **Evaluating server-assisted cache replacement in the Web**
Author(s): Cohen, E.; Krishnamurthy, B.; Rexford, J.
Author Affiliation: AT&T Bell Labs., Florham Park, NJ, USA
Book Title: Algorithms - ESA '98. 6th Annual European Symposium. Proceedings
Inclusive Page Numbers: 307-19
Publisher: Springer-Verlag, Berlin
Country of Publication: Germany
Publication Date: 1998
Conference Title: Proceedings of 6th Annual European Symposium on Algorithms
Conference Date: 24-26 Aug. 1998
Conference Location: Venice, Italy
Editor(s): Bilardi, G.; Italiano, G.F.; Pietracaprina, A. ; Pucci, G.
ISBN: 3 540 64848 8
Number of Pages: xii+513
Language: English
Document Type: Conference Paper (PA)
Treatment: Application (A); Practical (P)
Abstract: To reduce user-perceived latency in retrieving documents on the world wide web, a commonly used technique is caching both at the client's browser and more gainfully (due to sharing) at a proxy. The effectiveness of Web caching hinges on the replacement policy that determines the relative value of caching different objects. An important component of such policy is to predict next-request times. We propose a caching policy utilizing statistics on resource inter-request times. Such statistics can be collected either locally or at the server, and piggybacked to the proxy. **Using** various Web server **logs**, we compared existing cache replacement policies with our server-assisted schemes. The experiments show that utilising the server knowledge of access patterns can greatly improve the effectiveness of proxy caches. Our experimental evaluation and proposed policies use a price function framework. The price function values the utility of a unit of cache storage as a function of time. Instead of the usual tradeoffs of profit (combined value of **cache** hits) and **cache** size we measure tradeoffs of profit and caching cost (average allocated **cache portion**). The price-function framework allows us to evaluate and compare different replacement policies by **using** server **logs**, without having to construct a full workload model for each client's cache (*17 refs.*)
Subfile(s): C (Computing & Control Engineering)
Descriptors: cache storage; information resources; information retrieval
Identifiers: server-assisted cache replacement; user-perceived latency; world wide web; documents retrieval; caching; resource inter-request times; Web server logs; price-function framework
Classification Codes: C7250 (Information storage and retrieval); C7210N (Information